

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ БІЗНЕСУ ТА СУЧАСНИХ
ТЕХНОЛОГІЙ**

**ФОРМА НАВЧАННЯ ДЕННА
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ
ІНФОРМАТИКИ**

Допускається до захисту

Завідувач кафедри _____ О.О. Ємець
(підпис)

«_____» _____ 2020 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

ДО БАКАЛАВРСЬКОЇ РОБОТИ

на тему

**РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З ТЕМИ «РОЗВ'ЯЗУВАННЯ
ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ ПЕРШОГО ПОРЯДКУ» ДИСТАНЦІЙНОГО
НАВЧАЛЬНОГО КУРСУ
«МАТЕМАТИЧНИЙ АНАЛІЗ»**

зі спеціальності 122 «Комп'ютерні науки»

Виконавець роботи Ткаленко Юрій Олександрович

_____ «___» _____ 2021р.
(підпис)

Науковий керівник доц., каф.-мат. наук, Чілікіна Т.В.

_____ «___» _____ 2021р.
(підпис)

ПОЛТАВА 2021р.

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»

ЗАТВЕРДЖУЮ

Завідувач кафедри _____ **О.О. Ємець**

«8» вересня 2020р.

**Завдання та календарний графік
виконання дипломної роботи**

**Студент(ка) спеціальності 122 «Комп'ютерні науки»
Прізвище, ім'я, по батькові Ткаленко Юрій Олександрович**

1. Тема **«Розробка програмного забезпечення з теми «Розв'язування диференціальних рівнянь першого порядку»»** затверджена наказом ректора № 121-Н від «1» вересня 2020 р.

Термін подання студентом бакалаврської роботи «20» травня 2021 р.

2. Вихідні дані до дипломної роботи: публікації з теми навчальні тренажери в дистанційних курсах з комп'ютерних наук.

3. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

ВСТУП

1 ПОСТАНОВКА ЗАДАЧІ

1.1 Постановка задачі розробки тренажера

2 ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Огляд тренажерів, де реалізовано подібні завдання.

2.2. Переваги та вади розглянутих тренажерів.

2.3. Актуальність розробки програмного забезпечення

3 ТЕОРЕТИЧНА ЧАСТИНА

3.1 Загальні відомості

3.1.1 Диференціальні рівняння першого порядку

3.1.2 Веб-додаток

3.1.3 HTML та CSS

3.1.4 Мова програмування Javascript

3.1.5 Angular фреймворк

3.2 Алгоритм роботи тренажера

3.3 Блок-схема проходження тесту

4 ПРАКТИЧНА ЧАСТИНА

4.1 Опис програмного забезпечення

4.2 Складові частини веб-додатка

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТОК А

4. Перелік графічного матеріалу: 1 аркуш блок-схем, інші необхідні ілюстрації.

5. Консультанти розділів бакалаврської роботи

Розділ	Прізвище, ініціали, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1. Постанова задачі	Чілікіна Т.В.	8.09.20	8.09.20
2. Інформаційний огляд	Чілікіна Т.В.	8.09.20	8.09.20
3. Теоретична частина	Чілікіна Т.В.	8.09.20	8.09.20
4. Практична реалізація	Чілікіна Т.В.	8.09.20	8.09.20

6. Календарний графік виконання бакалаврської роботи

Зміст роботи	Термін виконання	Фактичне виконання
1. Вступ	10.05.21	
2. Вивчення методичних рекомендацій та стандартів та звіт керівнику	15.09.20	
3. Постановка задачі	1.10.20	
4. Інформаційний огляд джерел бібліотек та інтернету	2.11.20	
5. Теоретична частина	1.02.21	
6. Практична частина	17.05.21	
7. Закінчення оформлення	21.05.21	
8. Доповідь студента на кафедрі	28.05.21	
9. Доробка (за необхідністю), рецензування	14.06.21	

Дата видачі завдання « 8 » вересня 2020 р.

Студент Ткаленко Юрій Олександрович

Науковий керівник _____ доц., каф.-мат. наук, Чілікіна Т.В.
(підпис)

Результати захисту бакалаврської роботи

Дипломна робота оцінена на _____
(балів, оцінка за національною шкалою, оцінка за ECTS)

Протокол засідання ЕК № _____ від « _____ » _____ 2021 р.

Секретар ЕК _____
(підпис) (ініціали та прізвище)

РЕФЕРАТ

Записка: 42 стор., основна частина 38 стор., джерел - 10.

Предмет розробки – Розробка елементів програмного забезпечення у вигляді веб-додатку для навчання темі «Розв'язування диференціальних рівнянь першого порядку». Для створення програми було використано Angular Framework та середовище розробки PhpStorm.

Мета роботи – створити програмне забезпечення у вигляді веб-додатку для навчання студентів темі «Розв'язування диференціальних рівнянь першого порядку» дисципліни «Математичний аналіз»

Методи, які були використані для розв'язування задачі – Для створення програми було використано мову програмування Javascript, Angular фреймворк та програму PhpStorm.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	5
ВСТУП.....	5
1. ПОСТАНОВКА ЗАДАЧІ	6
1.1 Постановка задачі розробки тренажера.....	6
2. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	7
2.1. Огляд тренажерів, де реалізовано подібні завдання.	7
2.2. Переваги та вади розглянутих тренажерів	11
2.4. Актуальність розробки програмного забезпечення.....	12
3. ТЕОРЕТИЧНА ЧАСТИНА	15
3.1 Загальні відомості	15
3.1.1 Диференціальні рівняння першого порядку.....	15
3.1.2 Веб-додаток.....	17
3.1.3 HTML та CSS	18
3.1.4 Мова програмування Javascript.....	21
3.1.5 Angular фреймворк	22
3.2 Алгоритм роботи тренажера	23
3.3 Блок-схема проходження тесту	26
4. ПРАКТИЧНА ЧАСТИНА.....	27
4.1 Опис програмного забезпечення.....	27
ВИСНОВКИ.....	37
ДОДАТОК А.....	39

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

Умовні позначення, символи, скорочення, терміни	Пояснення умовних позначень, скорочень, символів
HTML (HyperText Markup Language)	мова розмітки гіпертексту
Фреймворк	інфраструктура програмних рішень, що полегшує розробку складних систем
CSS	це спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду
JavaScript (JS)	динамічна, об'єктно-орієнтована прототипна мова програмування

ВСТУП

Мета роботи – створити тренажер для навчання студентів темі "Розв'язування диференціальних рівнянь першого порядку" з дисципліни «Математичний аналіз»

Об'єкт роботи – створення програмного забезпечення у вигляді веб-додатку для систем дистанційного навчання.

Предмет роботи – програмне забезпечення для навчання темі "Розв'язування диференціальних рівнянь першого порядку"

Методи роботи – Для створення програми було використано мову програмування javascript і середовище розробки PhpStorm.

Структура пояснювальної записки до бакалаврської роботи:

- титульний аркуш;
- завдання до бакалаврської роботи;
- реферат;
- зміст;
- перелік умовних позначень, символів, одиниць, скорочень і термінів;
- вступ;
- теоретична частина;
- практична частина;
- висновки;
- список використаних джерел;

Обсяг пояснювальної записки: 42 стор., в т.ч. основна частина 38 стор., джерел

- 10.

1. ПОСТАНОВКА ЗАДАЧІ

1.1 Постановка задачі розробки тренажера

Основною задачею для дипломної роботи є програмна реалізація веб-додатка з теми «Розв'язування диференціальних рівнянь першого порядку»

Завданням для дипломної роботи є:

1. Розглянути актуальність теми;
2. Розглянути теоретичні матеріали про диференціальні рівняння першого порядку
3. Розробити алгоритм роботи тренажера;
4. Скласти блок-схему алгоритму роботи тренажера;
5. Програмна реалізація тренажера;
6. Опис роботи розробленого тренажера;
7. Тестування тренажера та перевірка на зручність для користувача

Основними вимогами до програмного забезпечення є:

1. Подача інформації у зрозумілому вигляді;
2. Зручний інтерфейс;
3. Перегляд кількості вірних відповідей;
4. Можливість перегляду кроків тренажера через довідку;
5. Повідомлення користувача про помилку або не вибрані елементи тесту;

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Огляд тренажерів, де реалізовано подібні завдання.

Тренажерів, для вивчення диференціальних рівнянь не було знайдено. Тому для огляду було розглянуто тренажер, що також був реалізований у вигляді веб-додатку з теми «Алгебра предикатів» дистанційного навчального курсу «Математична логіка»

На головній сторінці даного тренажеру відображається тема та є можливість переглянути теорію, або продовжити. Також є можливість вибору на якій мові продовжити (рис.2.1).



Рисунок 2.1 – Головна сторінка тренажеру з теми «Алгебра предикатів»
При переході на сторінку теорія відображається (рис.2.2).

Приклади алгебри предикатів

Приклад 1.

Навести приклади інтерпретацій формули алгебри предикатів $\forall x(P(x)) \rightarrow Q(x)$.

Розв'язання.

1. Область інтерпретації – множина живих істот, $P(x)$: x – риба, $Q(x)$: x живе у воді. Інтерпретація формули: «Всі риби живуть у воді».
2. Область інтерпретації – множина живих істот, $P(x)$: x – людина, $Q(x)$: x смертний. Інтерпретація формули: «Всі люди смертні».
3. Область інтерпретації – множина цілих чисел, $P(x)$: x ділиться на 6, $Q(x)$: x ділиться на 3. Інтерпретація формули: «Всі числа, які діляться на 6, діляться на 3».

Приклад 2.

Нехай предикат $P(x)$ відповідає реченню « $x \geq 1$ », а предметна область складається з усіх дійсних чисел. Тоді висловлювання $\forall x(P(x))$ хибне: $\forall x(P(x)) = F$.

Якщо ж предмета область складається з усіх натуральних чисел, то висловлювання $\forall x(P(x))$ істинне: $\forall x(P(x)) = T$.

Приклад 3.

Позначимо речення « x – просте число» як $P(x)$, « x – раціональне число» – як $Q(x)$, « x – дійсне число» як $R(x)$ та « x менше y » – як $МЕНШЕ(x, y)$. Розглянемо такі істинні твердження.

1. Кожне раціональне число дійсне.
2. Існує просте число.
3. Для кожного числа x існує таке число y , що $x < y$.

Наведені речення можна записати такими формулами.

1. $\forall x(P(x)) \rightarrow Q(x)$
2. $\exists x(P(x))$
3. $\forall x \exists y МЕНШЕ(x, y)$.

Рисунок 2.2 – Сторінка теорія тренажеру з теми «Алгебра предикатів»

При проходженні тесту відображаються варіанти для вибору (рис.2.3)

Алгебра предикатів

1

 Крок —

2

 Крок —

3

 Крок —

4

 Крок —

5

 Крок —

6

 Крок —

Нехай предикат $P(x)$ відповідає реченню « $x \geq 1$ ». Вказати предметну с

☐ Предметна область – множина дійсних чисел
 ☐ Предметна область – множина натуральних чисел
 ☐ Предметна область – множина цілих чисел
 ☐ Всі відповіді вірні

Спочатку

Далі

Рисунок 2.2 – Приклад запитання у тренажері з теми «Алгебра предикатів»

Якщо відповісти не вірно то з'явиться помилка, де буде відображена вірна відповідь (рис.2.4).

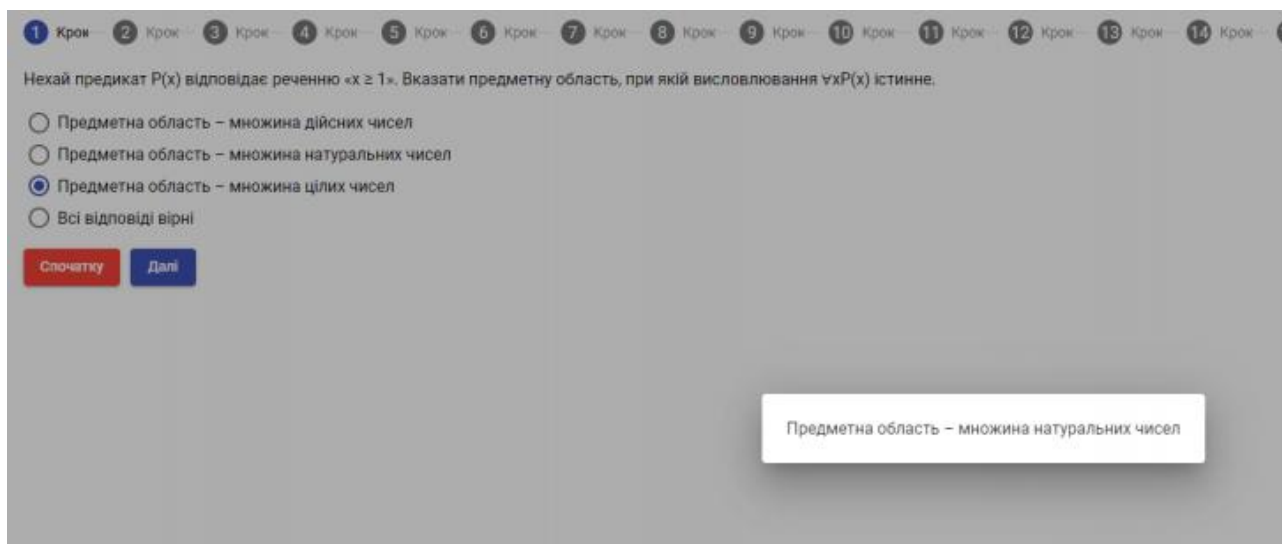


Рисунок 2.4 – Приклад помилки у тренажері з теми «Алгебра предикатів»

Якщо не відповісти і натиснути кнопку «Далі» то з'явиться помилка (рис.2.5).



Рисунок 2.5 – Повідомлення про необхідність вибрати відповідь у тренажері з теми «Алгебра предикатів»

При проходженні тесту повністю з'являється повідомлення про закінчення, та надається можливість почати тест спочатку (рис.2.6).

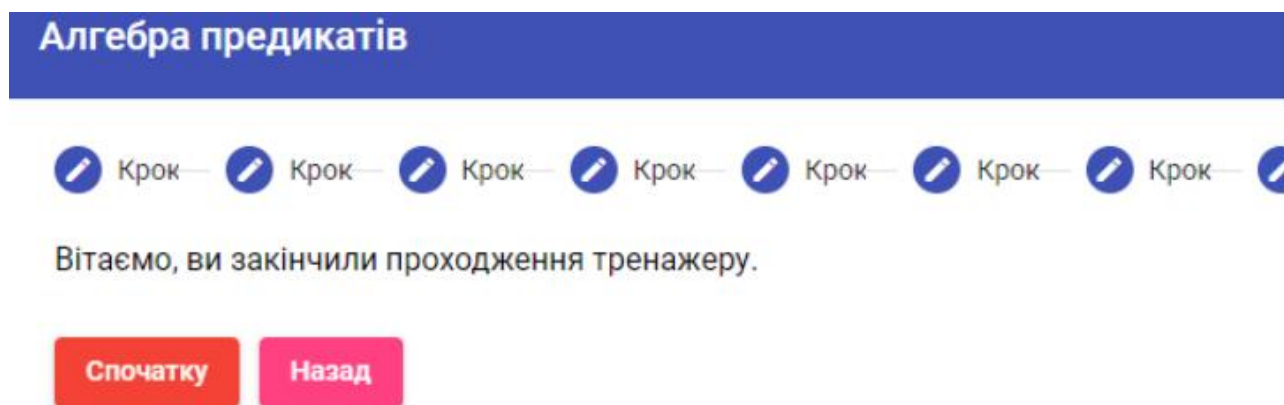


Рисунок 2.6 – Повідомлення про необхідність вибрати відповідь у тренажері з теми «Алгебра предикатів»

Якщо на початку вибрати англійську мову то весь контент на сайті буде відображатися на ній. Головна сторінка на англійській мові (рис2.7), сторінка «Теорія» на англійській мові (рис2.8).

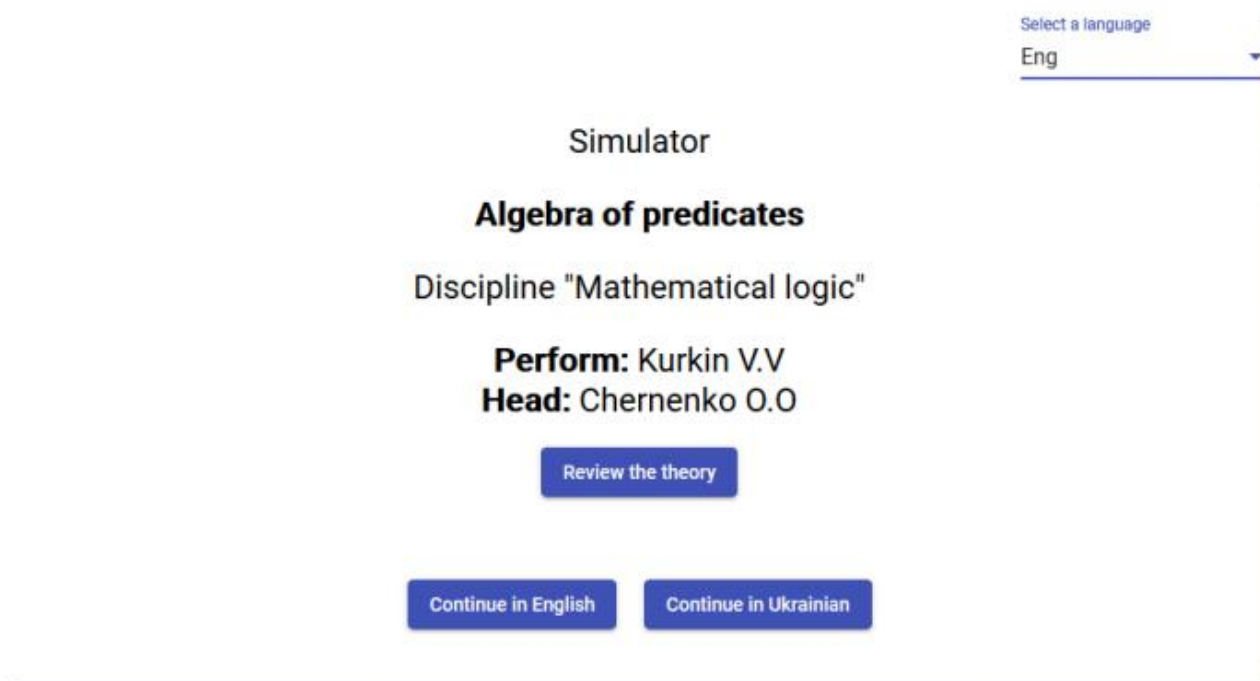


Рисунок 2.7 – Головна сторінка на англійській мові у тренажері з теми «Алгебра предикатів»

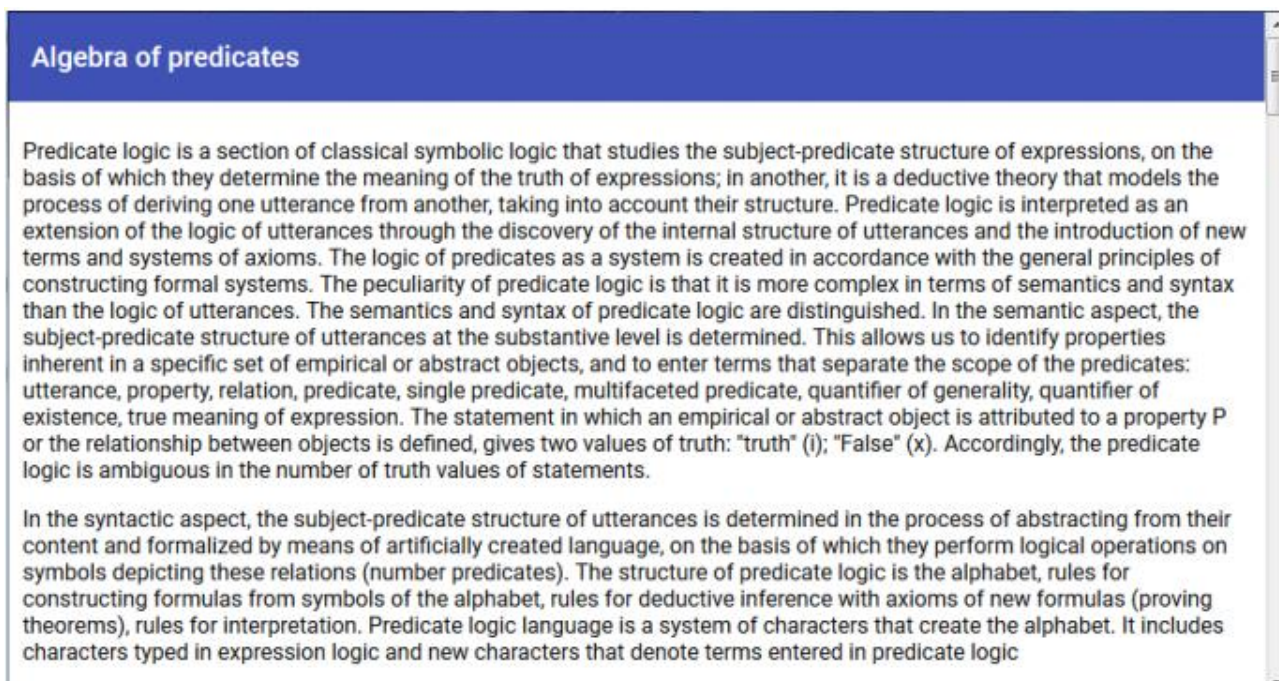


Рисунок 2.8 – Сторінка «Теорія» на англійській мові у тренажері з теми «Алгебра предикатів»

2.2. Переваги та вади розглянутих тренажерів

Перевагами можна назвати наступне:

1. Зручний інтерфейс.
2. Можливість вибору мови.
3. Наявність сторінки «Теорія», де можна ознайомитись з теоретичним матеріалом.
4. Перевірка вибраної відповіді.
5. Відображення кроків проходження тесту.

До недоліків можна віднести:

1. Помилка не виділяється кольором.
2. Відображення вірної відповіді у помилці, а не виділення вірної відповіді у віріантах відповіді.
3. Не має підрахунку кількості правильних відповідей.

2.4. Актуальність розробки програмного забезпечення

Оскільки зараз складна ситуація з можливістю відвідувати навчальний заклад, тому проекти, що дають змогу навчатися дистанційно зараз дуже важливі. Для більшого розуміння, можна додати інформацію про дистанційне навчання.

В основі дистанційної освіти закладені принципи та особливості традиційних форм навчання, окрім того, додані нові особливості, які полягають у використанні інтернет-технологій для доступу для навчальних матеріалів, інтерактивної взаємодії між студентами тощо.

Для ефективного дистанційного навчання важливим є принцип початкових знань, який полягає у тому, що користувач дистанційного курсу повинен володіти навичками роботи на комп'ютері, мати доступ до Інтернету, обов'язковими також є навички роботи в мережі та необхідне технічне забезпечення для повноцінного навчання.

Одним із найважливіших принципів у традиційній формі освіти є принцип науковості. Він ґрунтується на зв'язку між наукою та предметом, що вивчається, вимагає, щоб зміст матеріалів навчання забезпечував інформацію про наукові факти, поняття, закономірності, сучасні досягнення та відкриття. Більш фундаментальну форму цей принцип отримав у дистанційному навчанні: дидактичний принцип відповідності фундаментальності навчання пізнавальним потребам особи, яка навчається.

Цей принцип полягає у тому, що навчання вважається фундаментальним, якщо воно орієнтується на визначення основ та залежностей між різноманітними процесами навколишнього середовища.

Разом з цим, принцип висуває такі вимоги до студента, як:

- висока мотиваційна потреба;
- прагнення до власного розвитку;
- корекції та самокритики;
- спрямованість особистості до досягнення поставленої мети.

Продовженням змісту цього принципу є принцип вільного вибору інформації, яка отримується, шляхом визначеної діяльності: не існує єдиного ідеального інформаційного джерела, тому спрямованість навчання стосується безпосередньо не інформації, а шляхів її перетворення та опрацювання, за допомогою участі в дискусіях, телеконференціях, роботи з пошуковими машинами тощо.

Отже, з наведеної інформації можна зрозуміти, що дистанційне навчання забезпечує можливість навчатися тоді, коли зручно студенту, у тому темпі, що він сам обирає (в рамках установлених строків проведення курсів), в тому місці де він перебуває (не має потреби витратити час на дорогу до ВНЗ, для здачі поточних, а іноді і підсумкових контролів).

Варто зазначити, що дистанційне навчання в сучасному світі здійснюється за допомогою різноманітних технологій, які відрізняються за формою подання учбових матеріалів; наявністю посередника в системі навчання або за централізованою формою навчання; за ступенем використання телекомунікацій і персональних комп'ютерів.

За технологією організації контролю учбового процесу; за ступенем впровадження в технології навчання звичайних методів ведення освітнього процесу; за методами ідентифікації студентів при складанні іспитів.

Інформаційні технології повинні забезпечувати можливість контролю викладачем навчального процесу, можливість вносити зміни в навчальний курс, робити доступним контакт не лише викладача та 11 студента, а і забезпечувати можливість контактів студентів між собою – це принцип інтерактивності.

Тож дистанційні технології навчання можна розглядати як природний етап еволюції традиційної системи освіти від дошки з крейдою до електронної дошки й комп'ютерних навчальних систем, від книжкової бібліотеки до електронної, від звичайної аудиторії до віртуальної аудиторії

Отже, на базі даної інформації можна сказати, що розробка подібного програмного забезпечення є вкрай важливим, і потрібно рухатися в цьому напрямку.

Створення програмного забезпечення для навчання темі «Розв'язування диференціальних рівнянь першого порядку» є актуальним та потрібним.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1 Загальні відомості

Оскільки потрібно створити веб-додаток для навчання студентів темі "Розв'язування диференціальних рівнянь першого порядку" і було прийнято рішення для створення програми використовувати Angular Framework то основними елементами про які потрібно знати виступають поняття:

1. Диференціальні рівняння першого порядку;
2. Веб-додаток;
3. HTML та CSS
4. Мова програмування Javascript
5. Angular Framework

3.1.1 Диференціальні рівняння першого порядку

Диференціальні рівняння — рівняння, що встановлюють залежність між незалежними змінними, числами (параметрами), невідомими функціями та їхніми похідними. Невідома функція може бути як скалярною, так і векторною.

Рівняння вигляду

$$f\left(x, y, \frac{dy}{dx}, \dots, \frac{d^n y}{dx^n}\right) = 0 \quad (3.1)$$

називається диференціальним рівнянням (наявність похідних тут обов'язкова).

Найбільший порядок похідної, яка входить в диференціальне рівняння (3.1) називається порядком диференціального рівняння.

Функція $y(x)$ називається розв'язком (або інтегралом) диференціального рівняння (3.1), якщо вона n -раз неперервно диференційовна на деякому інтервалі $(a, b) = I$ і задовільняє диференціальному рівнянню (3.1) $\forall x \in I$.

$y'' + 3xy' + 2y = x^2$ - диференціальне рівняння другого порядку.

При $n=1$ диференціальне рівняння (2.1) називається диференціальним рівнянням першого порядку і позначається

$$F(x, y, y') = 0 \quad . \quad (3.2)$$

Диференціальне рівняння (2.2) називається розв'язаним відносно похідної, якщо його можна представити у вигляді

$$\frac{dy}{dx} = f(x, y) \quad . \quad (3.3)$$

Припускаємо, що $f(x, y)$ однозначна і неперервна в деякій області D змінних x, y . Цю область називають областю визначення диференціального рівняння (2.3).

Якщо в деякій області функція $f(x, y)$ перетворюється в ∞ , то розглядають диференціальне рівняння

$$\frac{dx}{dy} = \frac{1}{f(x, y)} \quad .$$

Множину таких точок, а також тих, в яких $f(x, y)$ не визначена, але може бути довизначена до неперервності, будемо приєднувати до області визначення диференціального рівняння (2.3).

Поряд з (2.3) будемо розглядати еквівалентне диференціальне рівняння, записане в диференціалах

$$dy - f(x, y)dx = 0, \quad (3.4)$$

або в більш загальному виді

$$M(x, y)dx + N(x, y)dy = 0 \quad (3.5)$$

Інколи розглядатимемо диференціальне рівняння в симетричній формі

$$\frac{dx}{X(x, y)} = \frac{dy}{Y(x, y)} \quad (3.6)$$

Функції $M(x, y)$, $N(x, y)$, $X(x, y)$, $Y(x, y)$ будемо вважати неперервними в деякій області.

Розв'язком диференціального рівняння (3.3) в інтервалі I назвемо функцію $y = \varphi(x)$, визначену і неперервно диференційовну на I , яка не

виходить з області означення функції $f(x, y)$ і яка перетворює диференціальне рівняння (2.3) в тотожність $\forall x \in I$, тобто

$$\frac{d\varphi(x)}{dx} \equiv f(x, y(x)), \quad x \in I$$

Розв'язок $y = \varphi(x)$ називається розв'язком, записаним в явній формі (вигляді).

Процес знаходження розв'язку диференціального рівняння називається інтегруванням.

Не завжди можна отримати розв'язок в явному вигляді.

Будемо говорити, що рівняння

$$\Phi(x, y) = 0 \tag{2.7}$$

визначає в неявній формі розв'язок диференціального рівняння (3.3), якщо воно визначає $y = y(x)$, яка є розв'язком диференціального рівняння (3.3).

При цьому на розв'язках диференціального рівняння (3.3) виконується

$$\Phi'_x(x, y) + \Phi'_y(x, y) \frac{dy}{dx} = \Phi'_x(x, y) + \Phi'_y(x, y) f(x, y) \equiv 0, \quad x \in I. \tag{3.8}$$

Будемо говорити, що співвідношення

$$x = \varphi(t), \quad y = \psi(t) \tag{3.9}$$

визначають розв'язок диференціального рівняння (2.3) в параметричній формі на інтервалі (t_0, t_1) , якщо

$$\frac{\psi'(t)}{\varphi'(t)} \equiv f(\varphi(t), \psi(t)), \quad t \in (t_0, t_1). \tag{3.10}$$

3.1.2 Веб-додаток

Веб-додаток - це будь-яка комп'ютерна програма, яка виконує певну функцію, використовуючи веб-браузер у якості свого клієнта. Додаток може бути таким же простим, як дошка оголошень або контактна форма на веб-сайті, або настільки ж складний, як текстовий процесор або багатокористувацький мобільний ігровий додаток, який ви завантажуєте на телефон.

Веб-додаток звільняє розробника від відповідальності за створення клієнта для певного типу комп'ютера або конкретної операційної системи, тому кожен може використовувати програму разом із наявним доступом до Інтернету. Оскільки клієнт працює у веб-браузері, користувач може використовувати IBM-сумісний або Mac. Вони можуть мати ОС Windows XP або Windows Vista. Вони навіть можуть використовувати Internet Explorer або Firefox, хоча для деяких програм потрібен певний веб-браузер.

Для розробки програми веб-додатки зазвичай використовують комбінацію сценарію на стороні сервера (ASP, PHP тощо) та сценарію на стороні клієнта (HTML, Javascript тощо). Сценарій на стороні клієнта займається поданням інформації, тоді як сценарій на стороні сервера займається всіма важкими речами, такими як зберігання та отримання інформації.

Протягом останніх декількох років спостерігається великий поштовх для розробки веб-додатків для функцій, яким зазвичай не потрібен сервер для зберігання інформації. Наприклад, ваш текстовий процесор зберігає документи на вашому комп'ютері та не потребує сервера.

Веб-додатки можуть забезпечити однакову функціональність та отримати перевагу роботи на кількох платформах. Наприклад, веб-додаток може діяти як текстовий процесор, зберігаючи інформацію в хмарі і дозволяючи "завантажувати" документ на свій особистий жорсткий диск.

3.1.3 HTML та CSS

Мова HTML (HyperText Markup Language – мова розмітки гіпертексту) являє собою стандартну мову, призначену для створення гіпертекстових документів в середовищі WWW (World Wide Web – Всесвітня павутина). HTML – документи (або Web-документи) можна переглядати веб-браузерами різних типів. Якщо документ створений з використанням HTML, веб-браузер може інтерпретувати HTML для виділення різних елементів документу і їх обробки.

Основною перевагою HTML являється те, що документ може бути переглянутий на веб-браузерах різних типів і на різних платформах. HTML — один із найбільш простих мов створення веб-сторінок.

HTML-документи можуть бути створені за допомогою любого текстового редактора чи спеціалізованих HTML-редакторів і конвертерів. HTML-документ — це звичайний файл у форматі ASCII.

На його основі лежать спеціалізовані дескриптори (теги), які і визначають правила форматування даних. Код програми написаний на мові HTML — звичайний текстовий файл, тому найпростіше його писати у простому Блокноті — стандартна Windows-програма. Код програми написаний в простому Блокноті стандартно зберігаються у форматі .txt, для того щоб зберегти це у вигляді Web-сторінки нам потрібно замість формату txt написати .html.

Ім'я HTML-документу слід писати латинськими літерами без пробілів. Також на підтримку мови HTML було створено серію мов програмування, таких як Java, JavaScript, Perl, PHP і т. д. З допомогою яких стало можливо створення динамічних веб-сторінок, включаючих форм, таблиці, фрейми та інші елементи.

Все, що є на веб-сторінці, створено за допомогою html-коду, який зберігається у файлах з типом html.

Кожне слово коду - спеціальна команда, яка говорить броузерам, як показувати веб-сторінку - де, що і якого вигляду на ній є.

Кожна команда мови html - це тег. Є парні і непарні теги.

Тег обрамлений кутовими дужками:

починається знаком менше < і закінчується знаком більше > .

Якщо тег парний, то закриваючий тег після знаку менше має слеш /.

Всередині тегу, крім його назви, можуть бути атрибути - тобто його властивості - із певними значеннями. Значення атрибуту пишеться у подвійних лапках після знака = після назви атрибуту.

Зараз HTML в чистому вигляді має дуже обмежений набір інструментів, що не дозволяє вирішувати ті чи інші дизайнерські та функціональні замисли

веб-ремісників. Ну ось хоч би, до прикладу, взяти початкове запитання всіх веб-ремісників "Як прибрати підкреслення у посиланні?" Або "Як змінити стиль посилання, при наведенні на нього курсора?" За допомогою лише одного HTML такого зробити не вдасться!. А таких запитань безліч. Тут й приходить на допомогу CSS, який вирішує більшість завдань, що відносяться до стильового оформлення сторінки.

Одна з головних переваг використання CSS - це можливість розділити зміст сторінки від її оформлення. Таке розділення дозволило покращити сприйняття та доступність змісту, забезпечити більшу гнучкість та контроль за відображенням змісту в різних умовах, зробити зміст більш структурованим та простим, прибрати повторення та ін. Власне це ж і була основна мета створення цієї технології.

Правила в CSS працюють по каскаду (пріорітету, вазі). Це дозволяє отримати передбачуваний результат у випадку, коли до одного елемента, одночасно, застосовуються декілька стильових правил.

Всі CSS-правила складаються з селектора і блоку оголошень (укладеного у фігурні дужки). В середині блоку оголошень може знаходитися одне або кілька оголошень, розділених крапкою з комою. Оголошення - це рядок, складений з css- властивості та її значення.

Кожне правило починається з селектора (показчика), що вказує на ті html-елементи, до яких ми будемо застосувати css-правило. У блоці оголошень відбувається найцікавіше - ми встановлюємо правила відображення обраних нами елементів, визначаємо їх властивості - розмір, колір, грані, поля, положення на екрані і т.д.

Отже, HTML використовується для структурування вмісту сторінки, а CSS - для форматування цього структурованого вмісту. А сукупності HTML та CSS дозволяють творити з документом неймовірні речі і поступово вивчаючи CSS ти почнеш переконуватися в цьому. Інколи створити деякий функціонал, набагато простіше за допомогою CSS аніж на JavaScript.

3.1.4 Мова програмування Javascript

JavaScript (скорочено JS) — динамічна, об'єктно-орієнтована мова програмування . Реалізація стандарту ECMAScript.

JavaScript найчастіше використовується у браузері , що надає можливість виконання коду на стороні клієнта (такому, що виконується на пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером , змінювати структуру та зовнішній вигляд веб-сторінки.

З появою Node.js мова програмування JavaScript також використовується для програмування на стороні сервера (подібно до таких мов програмування, як Java , PHP,), розробки ігор , стаціонарних та мобільних додатків, сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite), всередині PDF -документів тощо.

JavaScript має C-подібний синтаксис, але в порівнянні з мовою Сі має такі корінні відмінності:

- об'єкти, з можливістю інтроспекції і динамічної зміни типу через механізм прототипів функції як об'єкти першого класу обробка винятків
- автоматичне приведення типів
- автоматичне прибирання сміття
- анонімні функції

Не слід плутати Java і JavaScript. Мови Java та JavaScript є двома різними мовами, що мають відмінну семантику , хоча й мають схожі риси в стандартних бібліотеках та правилах іменування. Синтаксис обох мов отриманий «у спадок» від мови С , але семантика та дизайн JavaScript є результатом впливу мов Self та Scheme.

JavaScript містить декілька вбудованих об'єктів: Global , Object , Error, Function, Array, String, Boolean , Number , Math, Date , RegExp та інші.

JavaScript містить набір вбудованих операцій, які, строго кажучи, не обов'язково є функціями або методами, а також набір вбудованих операторів, що управляють логікою виконання програм.

JavaScript, наразі, є однією з найпопулярніших мов програмування в інтернеті. Але спочатку багато професіональних програмістів скептично ставилися до мови, цільова аудиторія якої складалася з програмістів-любителів. Поява AJAX змінила ситуацію та повернула увагу професійної спільноти до мови. В результаті, були розроблені та покращені багато практик використання JavaScript (зокрема, тестування та налагодження), створені бібліотеки та фреймворки, а з появою платформи Node.js поширилося використання JavaScript поза браузером.

3.1.5 Angular фреймворк

Фреймворк —інфраструктура програмних рішень, що полегшує розробку складних систем. Спрощено дану інфраструктуру можна вважати своєрідною комплексною бібліотекою, але при цьому вона має ряд обмежень, що задають правила створення структури проекту та написання коду.

На даний момент є багато фреймворків за допомогою яких можна реалізувати односторінковий додаток. У кожного є свої плюси і недоліки.

Angular - це клієнтський MVC-фреймворк, написаний на JavaScript. Він виконується в веб-браузері і надає величезну допомогу в створенні сучасних, односторінкових веб-додатків.

Angular привернув до себе увагу переважно завдяки своїй інноваційній системі шаблонів, простоті розробки з його використанням і застосування надійних інженерних рішень. його система шаблонів дійсно багато в чому унікальна:

- В якості мови шаблонів в ній використовується мова розмітки HTML.

- Вона не вимагає явно оновлювати дерево DOM, так як Angular здатний стежити за діями користувача, подіями браузера і змінами в моделі, і вчасно виявляти, коли і який шаблон потрібно оновити.
- Має вельми цікаву підсистему компонентів, і має можливість «навчати» браузер розпізнаванню і правильної інтерпретації нових тегів і атрибутів HTML.

3.2 Алгоритм роботи тренажера

Спочатку користувач потрапляє на сторінку з інформацією про тему даного веб-додатку (Рисунок 3.11).

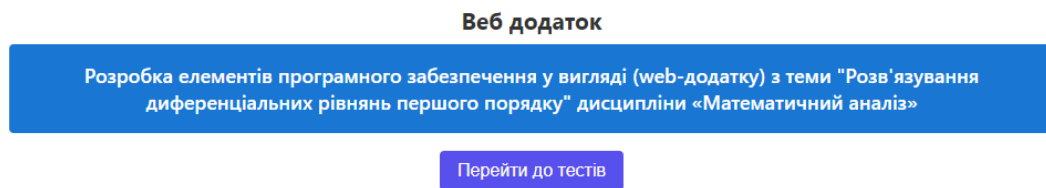


Рисунок 3.11 – Головна сторінка сайту

Далі користувач має можливість перейти на сторінку для проходження тесту (Рисунок 3.12). Щоб почати він повинен натиснути кнопку «Почати», тоді відобразиться запитання.

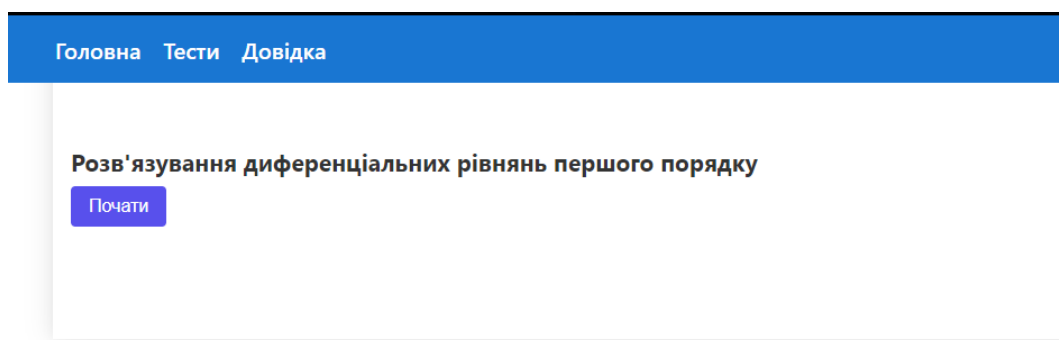


Рисунок 3.12 Тести

Користувач також має можливість перейти на сторінку “Довідка” (Рисунок 3.13) для отримання інформації про те, що робити для того щоб пройти тест.

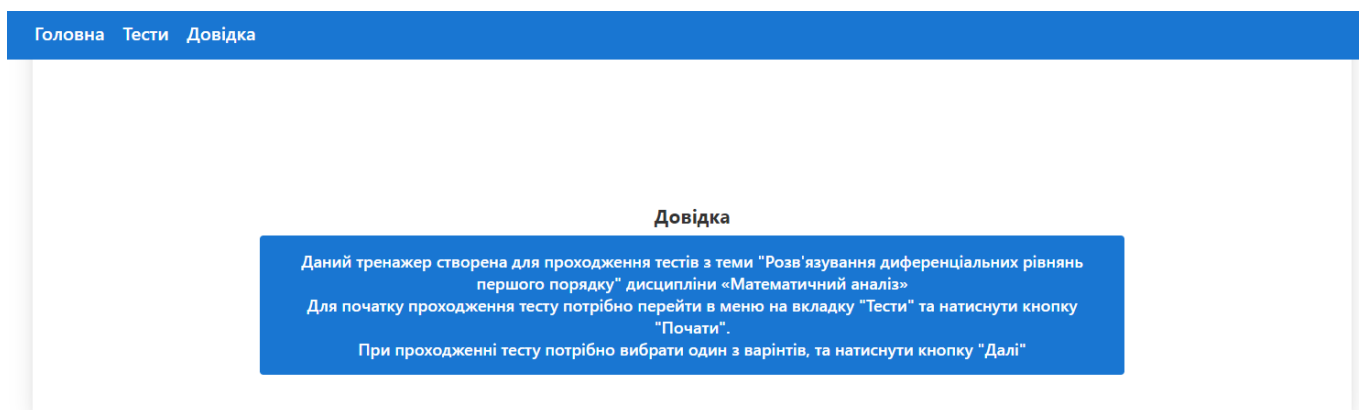


Рисунок 3.13 Довідка

Для проходження тесту користувачу потрібно натиснути на кнопку почати. Далі відкривається форма з показом запитання та варіантами відповіді (Рисунок 3.14).

Розв'язування диференціальних рівнянь першого порядку

Знайти загальний розв'язок диференціального рівняння.

- ☐ $y = 2x + 1$
- ☐ $y = x^2 + x$
- ☐ $y = x^2 + x + C$
- ☐ $y = 2x^2 + x + C$

Перевірити

Рисунок 3.14 Форма для проходження тесту

Алгоритм проходження тесту зображено на блок-схемі (Рисунок 3.15). На ньому зображено алгоритм, що дає зрозуміти всі можливі кроки під час проходження тесту.

Детальніше етапи проходження тесту буде розглянуто в практичній частині.

3.3 Блок-схема проходження тесту

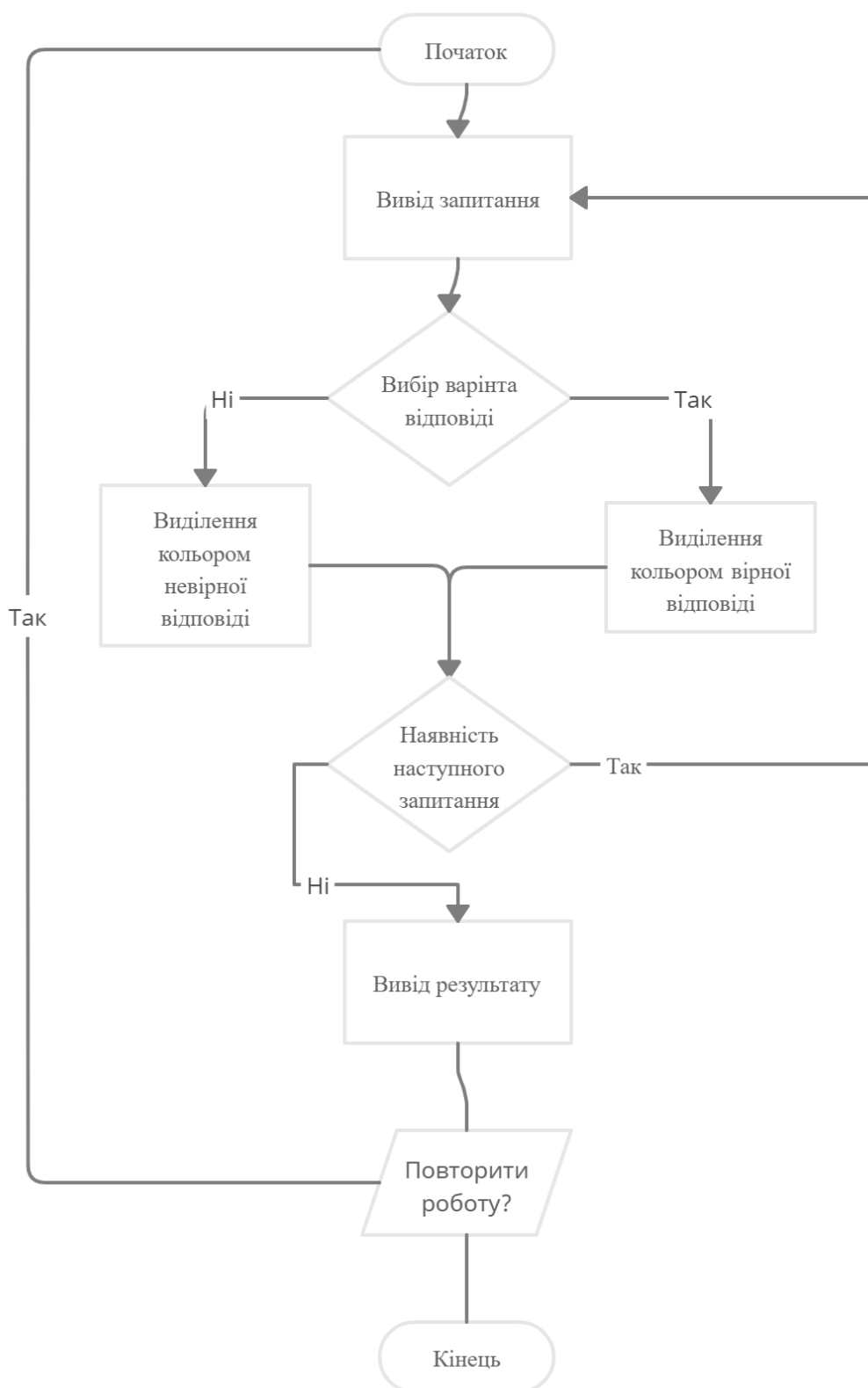


Рисунок 3.15 – Блок-схема проходження тесту

4. ПРАКТИЧНА ЧАСТИНА

4.1 Опис програмного забезпечення

JetBrains PhpStorm — комерційне крос-платформове інтегроване середовище розробки для PHP, яке розробляється компанією JetBrains на основі платформи IntelliJ IDEA.

PhpStorm являє собою інтелектуальний редактор для PHP, HTML і JavaScript з можливостями аналізу коду на льоту, запобігання помилок у сирцевому коді і автоматизованими засобами рефакторинга для PHP і JavaScript.

PhpStorm розроблений на основі платформи IntelliJ IDEA, написаної на Java. Користувачі можуть розширити функціональність середовища розробки за рахунок установки плагінів, розроблених для платформи IntelliJ, або написавши власні плагіни.

Вся функціональність WebStorm включена в PhpStorm.

4.2 Складові частини веб-додатка

Загалом сайт побудований з допомогою компонентів. В кожен з компонентів входять файли .html, .css, .ts (Рисунок 4.1).

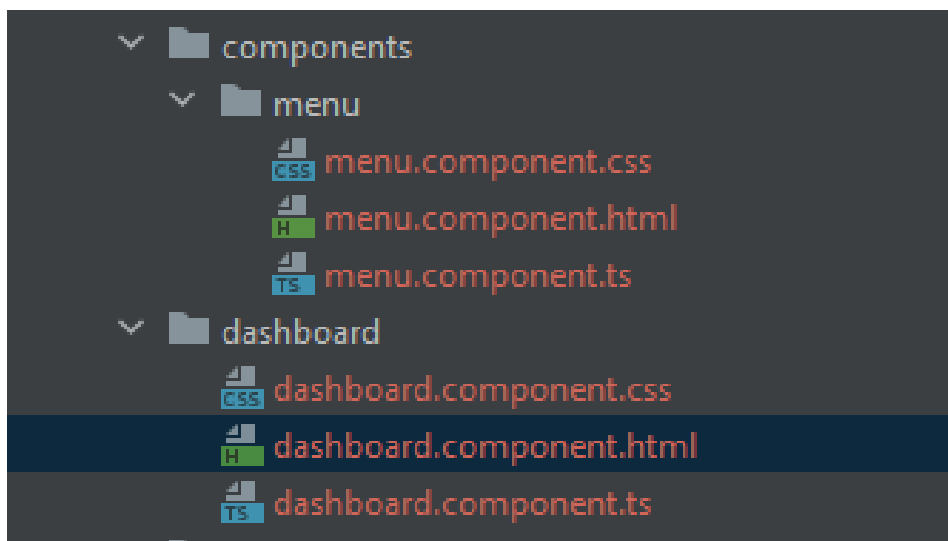


Рисунок 4.1 Структура компонентів

Спочатку для прикладу розглянемо компонент меню. На сайті меню має такий вигляд (Рисунок 4.2)

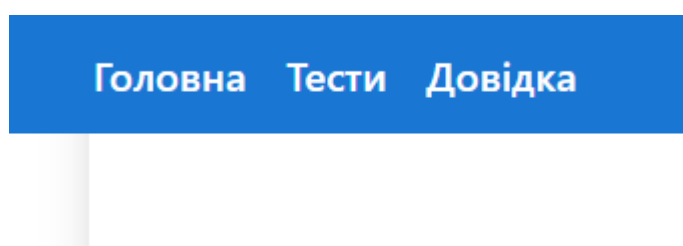


Рисунок 4.2 Меню.

Код для реалізації меню знаходиться у файлах .html (Рисунок 4.3) та .css (Рисунок 3.4). Оскільки Angular дає можливість додавати в .html файлах шлях на сторінку на яку потрібно буде потрапити по кліку, то в файлі .ts немає ніякого додаткового коду.

```

<div class="toolbar" role="banner">
  <div class="flex menu">
    <div class="menu-item">
      <a routerLink="/main">Головна</a>
    </div>
    <div class="menu-item">
      <a routerLink="/dashboard">Тести</a>
    </div>
    <div class="menu-item">
      <a routerLink="/help">Довідка</a>
    </div>
  </div>
</div>

```

Рисунок 4.3 Меню html.

На даному рисунку можна побачити також приклад використання елементів «RouterLink» для навігації по сайту.

```

.toolbar {
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  height: 60px;
  display: flex;
  align-items: center;
  background-color: #1976d2;
  color: white;
  font-weight: 600;
}

.toolbar .menu {
  max-width: 1440px;
  width: calc(100% - 64px);
  margin: 0 32px;
  display: flex;
}

.menu-item {
  color: white;
  font-size: 20px;
  margin: 10px;
}

.menu-item a {
  color: white;
  text-decoration: none;
}

.menu-item a:hover {
  text-decoration: underline;
}

```

Рисунок 4.4 Меню css.

Я не буду детально зупинятися на кожному файлі в кодї, оскільки не бачу в цьому необхідності, тому перейдемо до файлів, що стосуються самого тесту.

При переході на сторінку тестів, спочатку користувач бачить заголовок та кнопку (Рисунок 4.5).

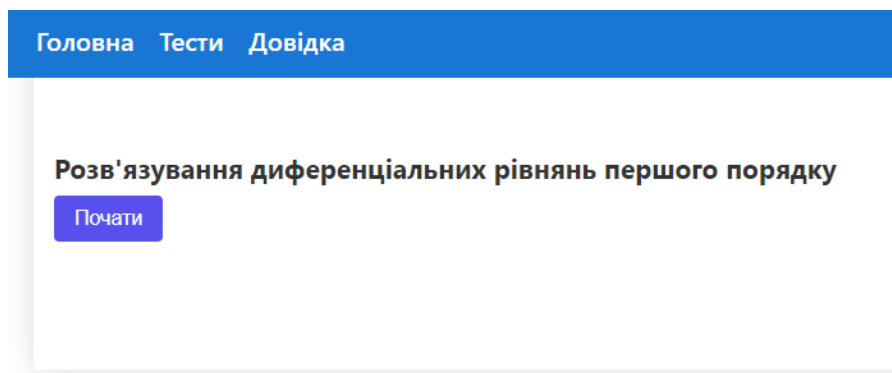


Рисунок 3.5 Тести початковий вигляд.

Далі при натисканні кнопки відображається перше питання тесту (Рисунок 4.6). Код для відображення тесту (Рисунок 4.7), (Рисунок 4.8),.

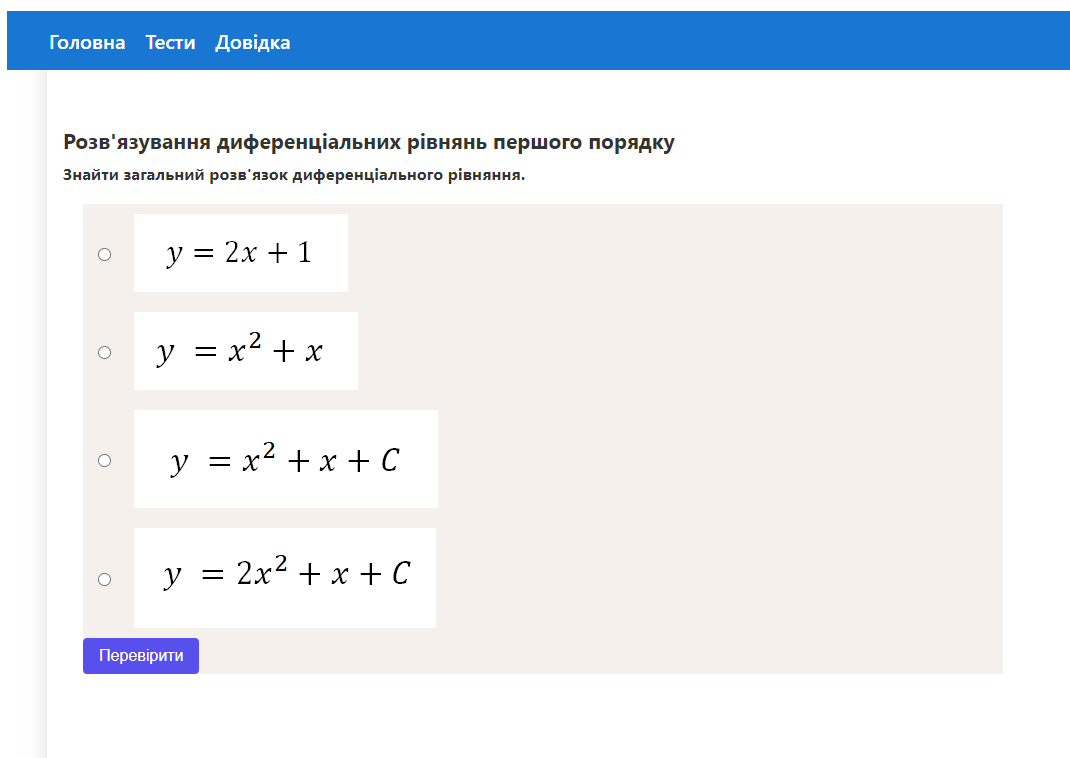


Рисунок 4.6 Тест.

```

<ng-template #test>
  <ng-container *ngIf="showTest; else showButton">
    <h3>{{selectedTest.title}}</h3>
    <div class="test-item">
      <ng-container *ngFor="let el of selectedTest.items">
        <div class="test-element"
          [ngClass]="{'success': el.success && checked , 'error': el.error && checked}">
          <label (click)="select(el)">
            <input type="radio" [value]="el.id" [name]="el.id"/>
            <img [src]="el.image" alt="">
          </label>
        </div>
      </ng-container>
      <ng-container *ngIf="!checked; else next">
        <ng-container *ngIf="showError">
          <h2 class="error">Ви повинні вибрати одне з значень</h2>
        </ng-container>
        <button (click)="checkSelected()">
          Перевірити
        </button>
      </ng-container>
      <ng-template #next>
        <button (click)="nextEv()">
          Далі
        </button>
      </ng-template>
    </div>
  </ng-container>
  <ng-template #showButton>
    <button (click)="showTest = true">
      Почати
    </button>
  </ng-template>
</ng-template>

```

Рисунок 4.7 Тест html.

```

.container {
  display: flex;
  flex-direction: column;
  flex: 1 1 auto;
  max-width: 1440px;
  width: calc(100% - 64px);
  margin: 0 32px;
  box-shadow: 0 0 25px 0 rgba(0, 0, 0, .1), 0 0 10px 0 rgba(0, 0, 0, .04);
}

.test-item {
  margin: 20px;
  background: #f5f0ec;
}

.test-element{
  padding: 10px;
}

.test-element.success {
  background: #00ef0073;
}

.test-element.error {
  background: #ff000085;
}

.test-element label {
  display: flex;
  align-items: center;
}

.test-element img{
  margin: 0 20px;
}

.error{
  color: #ff000085;
  margin: 10px;
}

```

Рисунок 4.8 Тест css.

Далі для проходження тесту користувач повинен вибрати один з елементів та натиснути кнопку «Перевірити». Далі проходить перевірка на те чи вибраний хоч один з елементів тесту, якщо ні – то виводиться повідомлення (Рисунок 4.9), якщо так - то йде перевірка на те чи правильний елемент вибрано.

Розв'язування диференціальних рівнянь першого порядку

Знайти загальний розв'язок диференціального рівняння.

- ☐ $y = 2x + 1$
- ☐ $y = x^2 + x$
- ☐ $y = x^2 + x + C$
- ☐ $y = 2x^2 + x + C$

Ви повинні вибрати одне з значень

Перевірити

Рисунок 4.9 Помилка про необхідність вибрати елемент.

Далі в залежності від того правильно вибрано елемент чи ні додається до вибраного елемента стилі, по яких легко зрозуміти правильний вибір чи ні. Відображення правильного варіанту вибору (Рисунок 4.10). Відображення неправильного варіанту вибору (Рисунок 4.11)

Розв'язування диференціальних рівнянь першого порядку

Знайти загальний розв'язок диференціального рівняння.

- ☐ $y = 2x + 1$
- ☐ $y = x^2 + x$
- ☒ $y = x^2 + x + C$
- ☐ $y = 2x^2 + x + C$

Далі

Рисунок 4.10 Вибрано вірний варіант.

Розв'язування диференціальних рівнянь першого порядку

Знайти загальний розв'язок диференціального рівняння.

- ☐ $y' = \cos 2x$
- ☐ $y = \frac{1}{2} \sin 2x$
- ☒ $y = 2 \sin 2x + C$
- ☐ $y = \frac{1}{2} \sin 2x + C$

Далі

Рисунок 4.11 Вибрано невірний варіант.

Перевірка правильності елемента визнається в залежності чи є у того елемента, що вибрано у об'єкті поле “isRight”. Сама структура тесту, що відображається виглядає так (Рисунок 4.12).

```
{
  id: 2,
  title: `Знайти загальний розв'язок диференціального рівняння.`,
  items: [
    {
      image: './assets/images/3-1.png',
      error: null,
      selected: null,
      success: null
    },
    {
      image: './assets/images/3-2.png',
      error: null,
      selected: null,
      success: null
    },
    {
      image: './assets/images/3-3.png',
      error: null,
      selected: null,
      isRight: true,
      success: null
    },
  ],
}
```

Рисунок 4.12 Структура тесту.

Коли користувач відповідає на останнє запитання йому відображається сторінка з підсумками тесту та з кнопкою «Пройти тест ще раз» (Рисунок 4.13). Вона дає можливість пройти тест ще раз.

Розв'язування диференціальних рівнянь першого порядку

Ви пройшли тест
Вірних відповідей: 3

[Пройти тест ще раз](#)

Рисунок 4.13 Підсумки тесту.

ВИСНОВКИ

Під час роботи над дипломною роботою було розроблено програмне забезпечення у вигляді веб-додатка для навчання темі "Розв'язування диференціальних рівнянь першого порядку" дисципліни «Математичний аналіз».

Результатами дипломної роботи є:

1. Вибрано методи роботи з програмним забезпеченням;
2. Виконано роботу над алгоритмом роботи веб-додатку;
3. Розроблено блок-схему проходження тесту;
4. Реалізоване програмне забезпечення у вигляді веб-додатку з допомогою Angular Framework та середовища розробки PhpStorm;
5. Розглянуто можливі варіанти помилок та опрацьовано їх.

Плюсами розробленого програмного забезпечення є:

1. Можливість відкрити веб-додаток у браузері;
2. Зручне подання інформації.

Основні мету та завдання дипломної роботи виконано, по результатах роботи створено програмне забезпечення у вигляді веб-додатку з теми "Розв'язування диференціальних рівнянь першого порядку" дисципліни «Математичний аналіз»

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Куркін В.В. Пояснювальна записка до дипломної роботи на тему програмне забезпечення для тренажера з теми «Алгебра предикатів» дистанційного навчального курсу «Математична логіка» / Куркін В.В. [Електронний ресурс].
Режим доступу до ресурсу: <http://dspace.puet.edu.ua/handle/123456789/7557>
2. Диференціальні рівняння [Електронний ресурс].
Режим доступу до ресурсу: <http://difur.in.ua/studentam/lektsii/lc03/>
3. Диференціальні рівняння [Електронний ресурс].
Режим доступу до ресурсу: <https://disted.edu.vn.ua/courses/learn/1283>
4. Angular (фреймворк) [Електронний ресурс].
Режим доступу до ресурсу:
[https://uk.wikipedia.org/wiki/Angular_\(%D1%84%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA\)](https://uk.wikipedia.org/wiki/Angular_(%D1%84%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA))
5. Angular (фреймворк) [Електронний ресурс].
Режим доступу до сайту: <https://webformyself.com/vvedenie-v-angular-chto-eto-za-frejmwork-i-zachem-ego-ispolzovat/>
6. Angular (фреймворк) [Електронний ресурс].
Режим доступу до ресурсу: https://itvdn.com/ru/blog/article/what_angular
7. Вивчення PhpStorm [Електронний ресурс].
Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/PhpStorm>
8. Що таке CSS [Електронний ресурс].
Режим доступу до ресурсу: https://css.in.ua/article/shcho-take-html_10
9. Javascript [Електронний ресурс].
Режим доступу до ресурсу: <http://xn--80adth0aefm3i.xn--j1amh/javascript>
- 10.Руководство по Angular 8 [Електронний ресурс].
Режим доступу до ресурсу: <https://metanit.com/web/angular2/>

ДОДАТОК А

```
{
  id: 0,
  title: `Серед зазначених рівнянь оберіть диференціальне.`,
  items: [{
    image: './assets/images/6-1.png',
    selected: null,
    error: null,
    success: null }, {
    image: './assets/images/6-2.png',
    error: null,
    selected: null,
    success: null }, {
    image: './assets/images/6-3.png',
    error: null,
    selected: null,
    success: null }, {
    image: './assets/images/6-4.png',
    error: null,
    isRight: true,
    selected: null,
    success: null
  ]
},
{
  id: 1,
  title: `Знайти загальний розв'язок диференціального рівняння.`,
  items: [
    {
      image: './assets/images/2-1.png',
      error: null,
      selected: null,
      success: null
    }
  ]
}
```

```

    },
    {
      image: './assets/images/2-2.png',
      error: null,
      selected: null,
      success: null
    },
    {
      image: './assets/images/2-3.png',
      error: null,
      selected: null,
      isRight: true,
      success: null
    },
    {
      image: './assets/images/2-4.png',
      error: null,
      selected: null,
      success: null] }},{
id: 2,
title: `Знайти загальний розв'язок диференціального рівняння.`,
items: [
  {
    image: './assets/images/3-1.png',
    error: null,
    selected: null,
    success: null
  },{
    image: './assets/images/3-2.png',
    error: null,
    selected: null,
    success: null
  }, {
    image: './assets/images/3-3.png',
    error: null,

```

```

    selected: null,
    isRight: true,
    success: null
  },
  {
    image: './assets/images/3-4.png',
    error: null,
    selected: null,
    success: null
  }
]
},
{
  id: 3,
  title: `Знайти загальний розв'язок диференціального рівняння`,
  items: [
    {
      image: './assets/images/4-1.png',
      selected: null,
      error: null,
      success: null
    }, {
      image: './assets/images/4-2.png',
      error: null,
      selected: null,
      success: null
    }, {
      image: './assets/images/4-3.png',
      error: null,
      selected: null,
      success: null
    }, {
      image: './assets/images/4-4.png',
      error: null,
      isRight: true,

```

```

    selected: null,
    success: null
  } ] },
{
  id: 4,
  title: `Знайти загальний розв'язок диференціального рівняння`,
  items: [
    {
      image: './assets/images/5-1.png',
      selected: null,
      error: null,
      success: null
    },
    {
      image: './assets/images/5-2.png',
      error: null,
      selected: null,
      success: null
    },
    {
      image: './assets/images/5-3.png',
      error: null,
      selected: null,
      success: null
    },
    {
      image: './assets/images/5-4.png',
      error: null,
      isRight: true,
      selected: null,
      success: null
    }
  ]
}

```